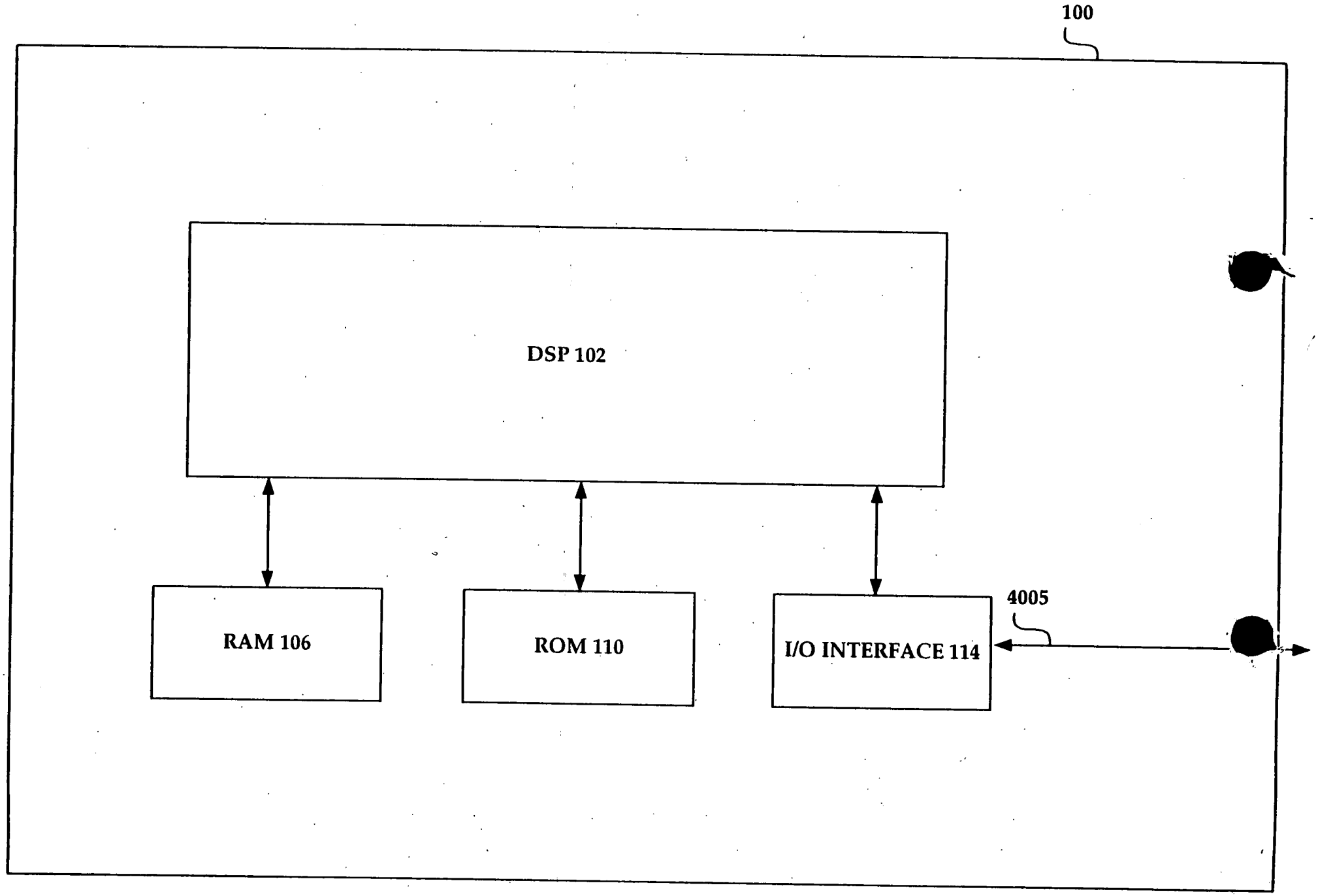

FIG. 1 (PRIOR ART)

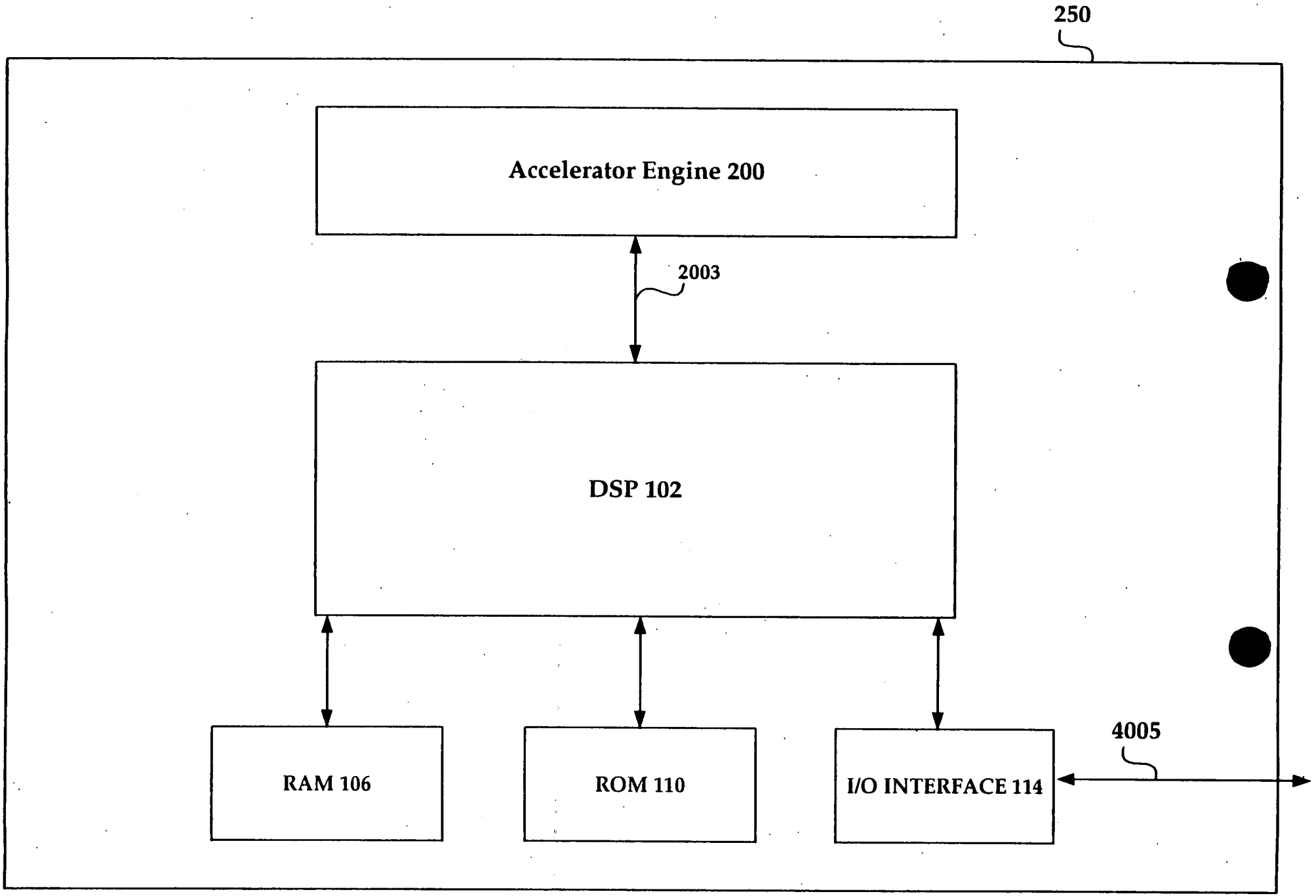

FIG. 2

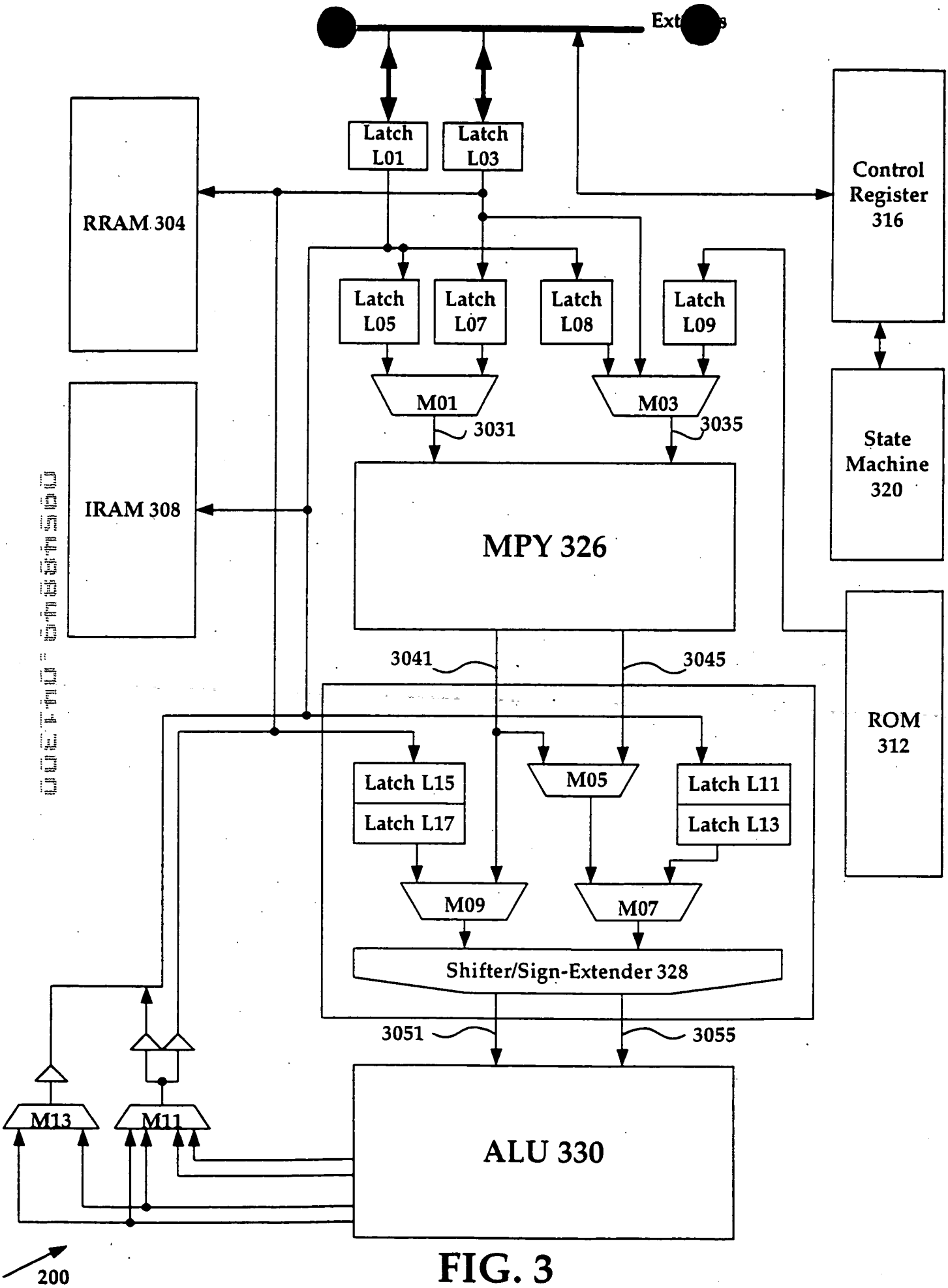
Latch L01
Latch L03
RRAM 304
Control Register 316
Latch L05
Latch L07
Latch L08
Latch L09
M01
M03
3031
3035
State Machine 320
IRAM 308
MPY 326
3041
3045
ROM 312
Latch L15
Latch L17
M05
Latch L11
Latch L13
M09
M07
Shifter/Sign-Extender 328
3051
3055
M13
M11
ALU 330
200

FIG. 3

Cycle	MPY 326	ALU A	ALU B	RRAM 304	IRAM 308	ROM 312
1				read b_r	read b_i	read c_r
2						read c_i
3	b_r*c_r					read c_i
4	b_i*c_i					read c_r
5	b_r*c_i					
6	b_i*c_r	$A=b_r*c_r$				
7		$A_0=A-(b_i*c_i)$				
8			$B=b_r*c_i$			
9			$B_0=B+(b_i*c_r)$			
10				write b_r	write b_i	
11						
12						

FIG. 4A

Cycle	MPY 326	ALU A	ALU B	RRAM 304	IRAM 308	ROM 312
1	1	0	1	1	1	1
2	1	1	0	1	1	1
3	1	1	0	0	0	1
4	1	0	1	0	0	1

FIG. 4B

Cycle	MPY 326	ALU A	ALU B	RRAM 304	IRAM 308	ROM 312
1				read b_r		read c_r
2					read b_i	read c_i
3	b_r*c_r					read c_i
4	b_i*c_i			read a_r		read c_r
5	b_r*c_i				read a_i	
6	b_i*c_r	$A=b_r*c_r$				
7		$A=A-b_i*c_i$				
8		$A_1=a_r-A$	$B=b_r*c_i$			
9		$A_0=a_r+A$	$B=B+(b_i*c_r)$			
10			$B_1=a_i-B$	write A_1		
11			$B_0=a_i+B$	write A_0	write B_1	
12					write B_0	

FIG. 5A

Cycle	MPY 326	ALU A	ALU B	RRAM 304	IRAM 308	ROM 312
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	1	1	1	1	1	1

FIG. 5B

```
Group = 1;
Block = FFT Length / 2;                          /* 64 or 32 */
R2P = Log (FFT Length) ;                         /* 7 or 6 */
for(i=0;i<R2P;i++)                               /* radix 2 pass counter */
        {
        Aiptr=0;                                 /* initialize A imaginary pointer */
        Arptr=0;                                 /* initialize A real pointer */
        Biptr=Block;                             /* initialize B imaginar pointer */
        Brptr=Block;                             /* initialize B real pointer */
        for(j=0;j<Group;j++)
                {
                for(k=0;k<Block;k++)
                        {
                        /* perform butterfly here */

                        ar = *Arptr;                        /* fetch data */
                        ai = *Aiptr;
                        br = *Brptr;
                        bi = *Biptr;

                        rtemp = br * cr – bi * ci;          /* perform complex multiply */
                        itemp = br * ci + bi * cr;

                        *Arptr++ = ar - rtemp;              /* update and write back data */
                        *Aiptr++ = ai – itemp;
                        *Brptr++ = ar + rtemp;
                        *Biptr++ = ai + itemp;
                        }
                Aiptr+=block;                       /* update addresses to next group */
                Arptr+=Block;
                Biptr+=Block;
                Brptr+=Block;
                }
        Block>>=1;                                  /* update block size for next radix 2 pass */
        Group<<=1;                                  /* update group size for next radix 2 pass */
        }
```

FIG. 5C

Cycle	MPY 326	ALU 330	RRAM 304	IRAM 308
1			read b_2	read x_{n-2}
2			read b_1	read x_{n-1}
3	b_2*x_{n-2}		read b_0	read x_n
4	b_1*x_{n-1}		read a_2	read y_{n-2}
5	b_0*x_n		read a_1	
6	a_2*y_{n-2}	$A=b_2*x_{n-2}$		
7	a_1*y_{n-1}	$A=A+(b_1*x_{n-1})$		
8		$A=A+(b_0*x_n)$		
9		$A=A+(a_2*y_{n-2})$		
10		$A_0=(A+(a_1*y_{n-1}))*2$		
11				
12				write y_n (A_0)

FIG. 6A

Cycle	MPY 326	ALU 330	RRAM 304	IRAM 308
1	1	1	1	1
2	0	1	1	1
3	1	1	1	1
4	1	1	1	1
5	1	0	1	0
6	1	1	0	1

FIG. 6B

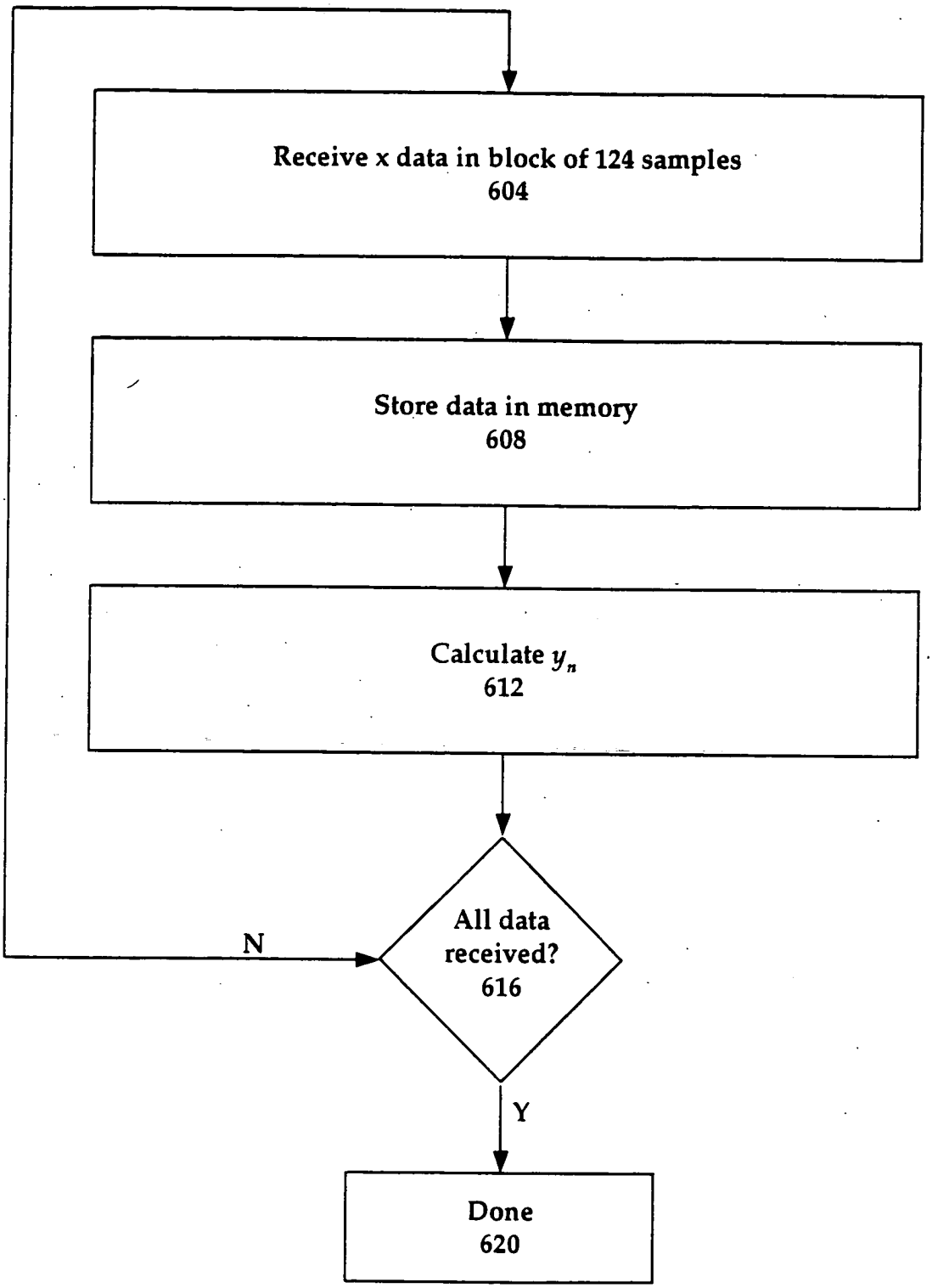
Receive x data in block of 124 samples
604
Store data in memory
608
Calculate y_n
612
All data received?
616
N
Y
Done
620

FIG. 6C

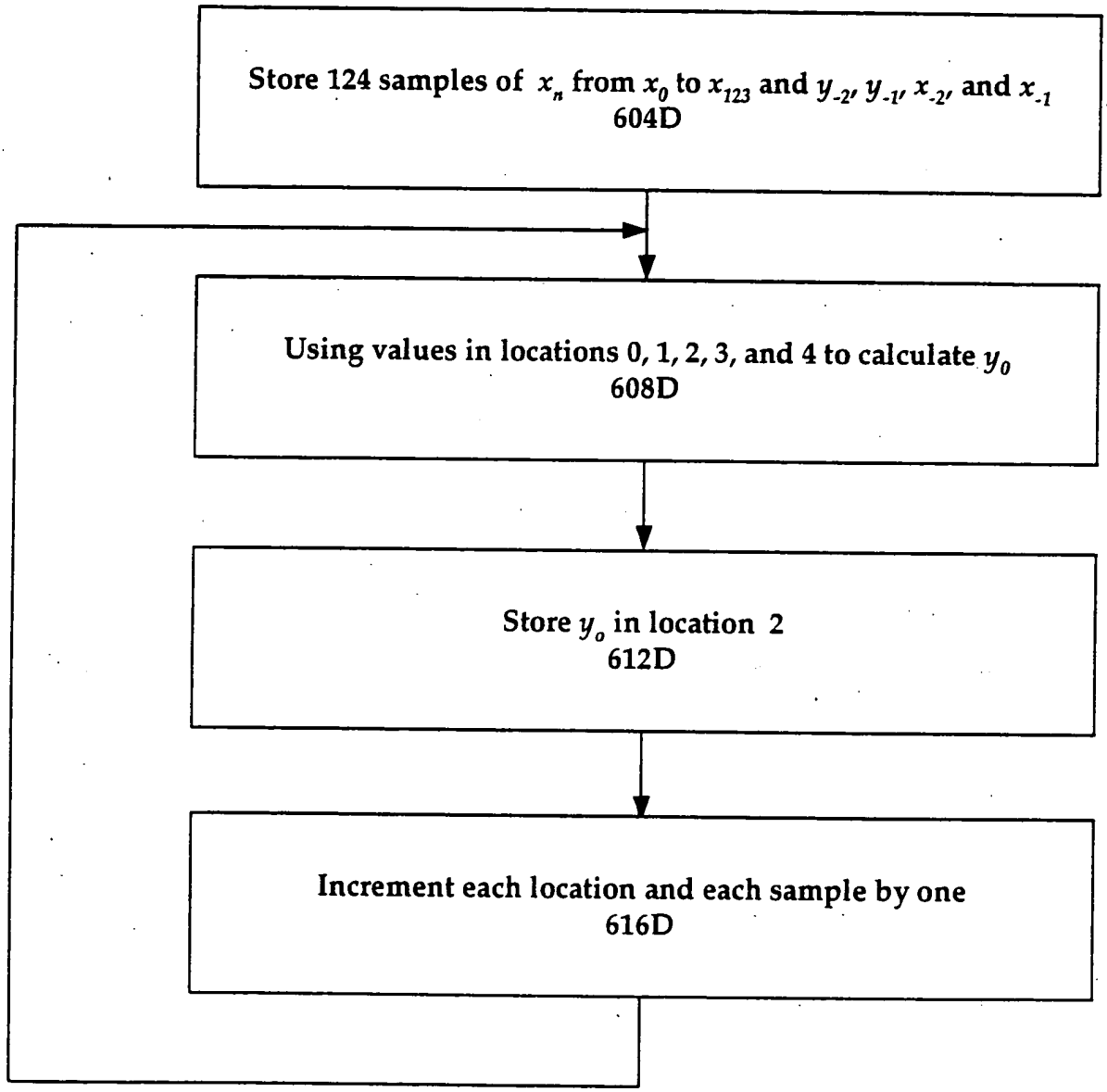
Store 124 samples of x_n from x_0 to x_{123} and y_{-2}, y_{-1}, x_{-2}, and x_{-1}
604D
Using values in locations 0, 1, 2, 3, and 4 to calculate y_0
608D
Store y_o in location 2
612D
Increment each location and each sample by one
616D

FIG. 6D

Address	Initial Data	Data n=0	Data n=1	Data n=2		Data n=123
0	y_{-2}	y_{-2}	y_{-2}	y_{-2}		y_{-2}
1	y_{-1} 655	y_{-1} 659	y_{-1} 663	y_{-1}		y_{-1}
2	x_{-2}	y_0	y_0	y_0 669		y_0
3	x_{-1}	x_{-1}	y_1	y_1		y_1
4	x_0	x_0 602	x_0	y_2	• • •	y_2
5	x_1	x_1	x_1 604	x_1		y_3
•	•	•	x_2	x_2		y_4
•	•	•	•	x_3		•
•	•	•	•	•		•
•	•	•	•	•		•
•	•	•	•	•		•
•	•	•	•	•		•
•	•	•	•	•		•
•	•	•	•	•		•
•	•	•	•	•		•
•	•	•	•	•		•
•	•	•	•	•		•
124	x_{120}	x_{120}	x_{120}	x_{120}		y_{122}
125	x_{121}	x_{121}	x_{121}	x_{121}		y_{123}
126	x_{122}	x_{122}	x_{122}	x_{122}		x_{122}
127	x_{123}	x_{123}	x_{123}	x_{123}		x_{123}

FIG. 6E

Cycle	MPY 326	ALU 330	RRAM 304	IRAM 308
1			read b_2	read x_{n-2}
2			read b_1	read x_{n-1}
3	b_2*x_{n-2}		read b_0	read x_n
4	b_1*x_{n-1}		read a_2	read yl_{n-2}
5	b_0*x_n			read y_{n-2}
6	a_2*yl_{n-2}	$A=b_2*x_{n-2}$	read a_1	
7	a_2*y_{n-2}	$A=A+(b_1*x_{n-1})$		
8	a_1*yl_{n-1}	$A=A+(b_0*x_n)$		
9	a_1*y_{n-1}	$A=A+(a_2*yl_{n-2})$		
10		$A=A+(a_2*y_{n-2})$		
11		$A=A+(a_1*yl_{n-1})$		
12		$A_0=A+(a_1*y_{n-1})$		
13				
14				
15				
16				
17				write yl_n (B_0)
16				write y_n (A_0)

FIG. 7A

Cycle	MPY 326	ALU 330	RRAM 304	IRAM 308
1	0	1	1	1
2	0	1	1	1
3	1	1	1	1
4	1	0	1	1
5	1	0	0	1
6	1	1	1	0
7	1	1	0	0
8	1	1	0	1
9	1	1	0	1

FIG. 7B

Address	Initial Data	Data n=0	Data n=1	Data n=2		Data n=121
0	yh_{-2}	yh_{-2}	yh_{-2}	yh_{-2}		yh_{-2}
1	yh_{-1} 755	yh_{-1} 759	yh_{-1} 763	yh_{-1}		yh_{-1}
2	yl_{-2}	yh_0	yh_0	yh_{-0} 769		yh_0
3	yl_{-1}	yl_{-1}	yh_1	yh_1		yh_1
4	x_{-2}	yl_0 702	yl_0 704	yh_2	• • •	yh_2
5	x_{-1}	x_{-1}	yl_1	yl_1		yh_3
•	x_0	x_0 703	x_0 705	yl_2		yh_4
•	•	x_1	x_1	x_1		yh_5
•	•	•	x_2	x2		•
•	•	•	•	x3		•
•	•	•	•	•		•
•	•	•	•	•		•
•	•	•	•	•		•
•	•	•	•	•		•
•	•	•	•	•		•
•	•	•	•	•		yh_{120}
•	•	•	•	•		yh_{121}
124	x_{118}	x_{118}	x_{118}	x_{118}		yh_{120}
125	x_{119}	x_{119}	x_{119}	x_{119}		yh_{121}
126	x_{120}	x_{120}	x_{120}	x_{120}		x_{120}
127	x_{121}	x_{121}	x_{121}	x_{121}		x_{121}

FIG. 7C

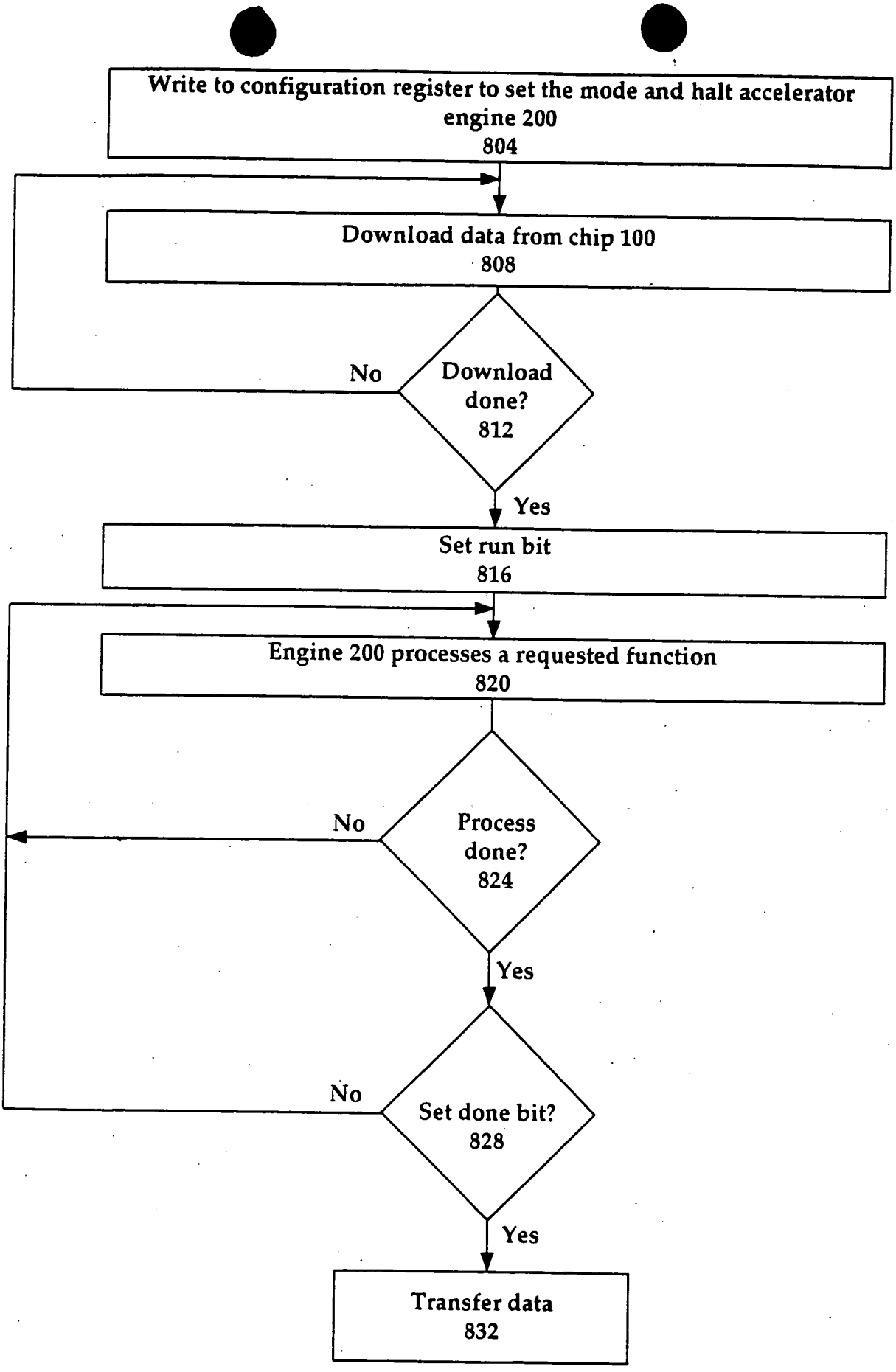
Write to configuration register to set the mode and halt accelerator engine 200
804
Download data from chip 100
808
Download done?
812
No
Yes
Set run bit
816
Engine 200 processes a requested function
820
Process done?
824
No
Yes
Set done bit?
828
No
Yes
Transfer data
832

FIG. 8